

plh



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/618,965	07/19/2000	Robert S. French	003048.P010	2144

23363 7590 06/29/2004

CHRISTIE, PARKER & HALE, LLP
PO BOX 7068
PASADENA, CA 91109-7068

EXAMINER

LE, DIEU MINH T

ART UNIT	PAPER NUMBER
2114	12

2114

DATE MAILED: 06/29/2004

12

Please find below and/or attached an Office communication concerning this application or proceeding.

PRE

Office Action Summary	Application 09/618,965	Applicant(s) FRENCH ET AL.	
	Examiner Dieu-Minh Le	Art Unit 2114	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 01 June 2004.
- 2a) This action is **FINAL**. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-80, 83 and 84 is/are pending in the application.
 - 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-80, 83 and 84 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 - 1. Certified copies of the priority documents have been received.
 - 2. Certified copies of the priority documents have been received in Application No. _____.
 - 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
 Paper No(s)/Mail Date _____
- 4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date. _____
- 5) Notice of Informal Patent Application (PTO-152)
- 6) Other: _____

Art Unit: 2114

1. This Office Action is a response to the RCE filed June 01, 2004 in application 09/618,965.

Part III DETAILED ACTION

Specification

1. Claims 1-80, 83 & 84 are again presented for examination; claims 81-82 have been canceled.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time

Art Unit: 2114

any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(f) or (g) prior art under 35 U.S.C. 103(a).

4. Claims 1-80, 83-84 are rejected under 35 U.S.C. § 103(a) as being unpatentable over You (US Patent 6,158,045) in view of Cardoza et al. (US Patent 5,630,049 hereafter referred to as Cardoza).

As per claim 1:

You substantially teaches the invention. You explicitly teaches:

- A method for interactive debugging debugging in a multi-channel, multi-service [abstract, col. 1, lines 18-30, col. 5, lines 47-55] comprising:

- dynamically allocating a plurality of services to a plurality of processors in the multi-channel, multi-service system for execution [col. 6, lines 20-30, and col. 87, lines 48-65];

- selecting a target construct from the dynamically allocated executing plurality of services for debugging [col. 1, lines 32-44, col. 6, lines 7-10, col. 7, lines 49-52, and col. 41, lines 35-55];

Art Unit: 2114

- accessing data related to an operation of the target construct by a debug [col. 6, lines 13-16 and col. 64, lines 30-33];

You does not explicitly address:

- monitoring at least a portion of data without disturbing the operation.

However, You does disclose capability of:

- an interactive debugging system and method in supporting the dynamic computer environment [col. 1, lines 18-30 8]

comprising capabilities of:

- inspecting the state of the program via debugging [col. 6, lines 17-30];

- inspecting and controlling the execution of the program for its uninterrupted operation [col. 6, lines 38-52].

- debugging services used for debugging can be executed in microprocessor, register sizes, processor register, memory addresses, operating system, hardware debugging support, operating system debugging support, software, multitude of parameters, etc... [col. 9, lines 16-26];

- data encapsulation in various of information, such as process state, execution state, etc... in supporting the interactive debugging [col. 10, lines 1-7].

In addition, Cardoza explicitly teaches:

- a apparatus and method for testing and debugging in a computer network [abstract, col. 1, lines 5-10] comprising:

- status information on portion of the target operating system code or data without disturbing the operation [col. 5, lines 3-60];

Art Unit: 2114

- an isolated debugging environment for the computer resources (i.e., services) [col. 2, lines 35-41].

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of Applicant's invention to implement, first, to realize the You' s interactive debugging system and method in supporting the dynamic computer environment comprising capabilities of debugging services used for debugging can be executed in microprocessor, register sizes, processor register, memory addresses, operating system, hardware debugging support, operating system debugging support, software, multitude of parameters, etc... for its uninterrupted operation as being as being the monitoring at least a portion of data without disturbing the operation as claimed by Applicant; since the You' s interactive debugging system and method in supporting the dynamic computer environment does perform the data tracking, checking, inspection (i.e., data monitoring), and isolating failure in supporting the continuity networking operation. It would have further obvious to an ordinary skill in the art to set-up checkpoint to inspecting or monitoring data or portion of data in ensuring the execution of the system performed properly; second, one would modify the You' s interactive debugging system and method in supporting the dynamic computer environment to explicitly including the status information on portion of the target operating system code or data without disturbing the operation as taught by Cardoza's apparatus and method for testing and debugging in a computer network in supporting the interactive debugging within a computer data networking environment.

Art Unit: 2114

This modification would have been obvious because a person having ordinary skill in the art would have been motivated to do so to provide the connectivity among data processor, memory, input/output device and other networking computer devices with mechanism to enhance the data processing, more specifically to interactive debugging in supporting the network establishment error detection/correction for maximizing the data process performance and throughput.

As per claim 2:

You substantially teaches the invention. You explicitly teaches:

- A method for interactive debugging [abstract, col. 1, lines 18-30, col. 5, lines 47-55] comprising:
 - modifying at least portion of data [col. 6, lines 38-42];
 - inspecting the state of the program via debugging [col. 6, lines 17-30].

As per claims 3-5:

You substantially teaches the invention. You explicitly teaches:

- A method for interactive debugging [abstract, col. 1, lines 18-30, col. 5, lines 47-55,] comprising:
 - a target construct for debugging [col. 1, lines 32-44, col. 6, lines 7-10, col. 7, lines 49-52, and col. 41, lines 35-55];
 - select from group of a service, a socket, a service stack, a set of service, and a set of sockets [col. 52,

Art Unit: 2114

lines 14-42, col. 56, lines 10-30, and col. 81, lines 26-33];

In addition, Cardoza explicitly teaches:

- a apparatus and method for testing and debugging in a computer network [abstract, col. 1, lines 5-10] comprising:
 - status information on portion of the target operating system code or data without disturbing the operation [col. 5, lines 3-60];
 - booting and polling services [col. 14, lines 30-40];
 - breakpoints and show call debugging services [col. 26, lines 18 through col. 27, lines 50].

As per claims 6-8:

You substantially teaches the invention. You explicitly teaches:

- A method for interactive debugging [abstract, col. 1, lines 18-30, col. 5, lines 47-55,] comprising:
 - a target construct for debugging [col. 1, lines 32-44, col. 6, lines 7-10, col. 7, lines 49-52, and col. 41, lines 35-55];
 - selecting the target construct from a plurality of sockets [col. 52, lines 14-42, col. 56, lines 10-30, and col. 81, lines 26-33];
 - information including a current state of services [col. 6, lines 45-52, col. 7, lines 28-40, and col. 75, lines 35-67].

In addition, Cardoza explicitly teaches:

- a apparatus and method for testing and debugging in a computer network [abstract, col. 1, lines 5-10] comprising:

Art Unit: 2114

- status information on portion of the target operating system code or data without disturbing the operation [col. 5, lines 3-60];
- booting and polling services [col. 14, lines 30-40];
- breakpoints and show call debugging services [col. 26, lines 18 through col. 27, lines 50].

As per claims 9-12:

You substantially teaches the invention. You explicitly teaches:

- A method for interactive debugging [abstract, col. 1, lines 18-30, col. 5, lines 47-55,] comprising:
 - a target construct for debugging [col. 1, lines 32-44, col. 6, lines 7-10, col. 7, lines 49-52, and col. 41, lines 35-55];
 - accessing corresponding to reading and writing to the memory [col. 6, lines 38 through col. 7, line 10, col. 56, lines 1-9];
 - modifying state [col. 6, lines 38-42];
 - dynamically allocating the debug construct [col. 6, lines 20-30, and col. 87, lines 48-65].

You does not explicitly address:

- dynamically de-allocation the debug construct once monitoring completed.

However, You does disclose capability of:

- an interactive debugging system and method in supporting the dynamic computer environment [col. 1, lines 18-30 8] comprising capabilities of :

Art Unit: 2114

- inspecting the state of the program via debugging [col. 6, lines 17-30];
- inspecting and controlling the execution of the program for its uninterrupted operation [col. 6, lines 38-52].
- dynamically allocating the debug construct [col. 6, lines 20-30, and col. 87, lines 48-65.

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of Applicant's invention to implement to realize the You' s interactive debugging system and method in supporting the dynamic computer environment comprising capabilities inspecting the state of the program via debugging, and inspecting and controlling the execution of the program for its uninterrupted as well as dynamically allocating the debug construct operation as being as being the dynamically de-allocation the debug construct once monitoring completed as claimed by Applicant since the You' s interactive debugging system and method in supporting the dynamic computer environment does perform the inspection and allocation of data in a dynamically debugging environment. That in turn, the interactive debugging system would have been dynamically de-allocation the debugging in ordering the system to ensuring the process performed properly. It would have been obvious to an ordinary skill in the art to apply and allocation and de-allocation of debugging process to achieve the data debugging method.

As per claims 14-18 and 21:

You substantially teaches the invention. You explicitly teaches:

Art Unit: 2114

- A method for interactive debugging [abstract, col. 1, lines 18-30, col. 5, lines 47-55,] comprising:
 - a target construct for debugging [col. 1, lines 32-44, col. 6, lines 7-10, col. 7, lines 49-52, and col. 41, lines 35-55];
 - transmitting data to host system [col. 6, lines 26-30 and lines 53-55];
 - transmitting based upon a request sent by a host application [col. 1, lines 38-45];
 - an operating system determines which data is to be transmitted [col.7, lines 56 through col. 8, lines 8];
 - debug construct specifies data is to be transmitted [col. 4, lines 45-50];
 - notifying the debug construct upon a completion of operation [col. 8, lines 29-44 and lines 55-67].
 - sending the request and transmitting the response are performed over a network [col. 1, lines 38-45].

As per claims 13 and 19:

You substantially teaches the invention. You explicitly teaches:

- A method for interactive debugging [abstract, col. 1, lines 18-30, col. 5, lines 47-55,] comprising:
 - a target construct for debugging [col. 1, lines 32-44, col. 6, lines 7-10, col. 7, lines 49-52, and col. 41, lines 35-55];
 - accessing corresponding to reading and writing to the memory [col. 6, lines 38 through col. 7, line 10, col. 56, lines 1-9];
 - modifying state [col. 6, lines 38-42];

Art Unit: 2114

- dynamically allocating the debug construct [col. 6, lines 20-30, and col. 87, lines 48-65].

You does not explicitly address:

- collecting statistic related to target and measuring bandwidth required to transmitted data.

However, You does disclose capability of:

- an interactive debugging system and method in supporting the dynamic computer environment [col. 1, lines 18-

30]comprising capabilities of:

- performance measurement for it data transmission bottlenecks in improving data quality [col. 6, lines 26-30];

- dynamically information gathering [col. 6, lines 6-30].

- dynamically allocating the debug construct [col. 6, lines 20-30, and col. 87, lines 48-65].

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of Applicant's invention to implement to realize the You' s interactive debugging system and method in supporting the dynamic computer environment comprising capabilities performance measurement for it data transmission bottlenecks in improving data quality and dynamically information gathering operation as being as being the collecting statistic related to target and measuring bandwidth required to transmitted data as claimed by Applicant since the You' s interactive debugging system and method in supporting the dynamic computer environment does deal with data access control, data gathering, data set breakpoint,

Art Unit: 2114

data targeting in supporting the interactive debugging system. Therefore it would have been obvious to an ordinary skill in the art to apply You's interactive debugging capabilities for performing the bandwidth measurement and collecting statistic related to target debugging process in ensuring the system operation performing uninterruptedly.

As per claims 20 and 22-23:

You substantially teaches the invention. You explicitly teaches:

- A method for interactive debugging [abstract, col. 1, lines 18-30, col. 5, lines 47-55] comprising:
 - a target construct for debugging [col. 1, lines 32-44, col. 6, lines 7-10, col. 7, lines 49-52, and col. 41, lines 35-55];
 - debugging performed in a multi-channel, multi-service environment [col. 8, lines 55-67 and col. 79, lines 20-24];
 - transmitting data to host system [col. 6, lines 26-30 and lines 53-55];
 - transmitting based upon a request sent by a host application [col. 1, lines 38-45];
 - an operating system determines which data is to be transmitted [col.7, lines 56 through col. 8, lines 8];
 - debug construct specifies data is to be transmitted [col. 4, lines 45-50];
 - allocating a copy of target construct in a simulated environment [col. 5, lines 61-67 and col. 41, lines 36-55].
 - collecting data [col. 1, lines 32-44, col. 6, lines 7-10, col. 7, lines 49-52, and col. 41, lines 35-55];

Art Unit: 2114

As per claim 24:

You substantially teaches the invention. You explicitly teaches:

- A method for multi-channel, multi-service debugging [col. 8, lines 55-67 and col. 79, lines 20-24] comprising:
 - providing information about service [col. 1, lines 32-44, col. 6, lines 7-10, col. 7, lines 49-52];
 - selecting a target construct for debugging [col. 1, lines 32-44, col. 6, lines 7-10, col. 7, lines 49-52, and col.41, lines 35-55];
 - dynamically loading one or more of the plurality of running services into the target construct [col. 5, lines 61-67 and col. 41, lines 36-55].

You does not explicitly address:

- maintaining an isolated debugging environment for service.

However, You does disclose capability of:

- an interactive debugging system and method in supporting the dynamic computer environment [col. 1, lines 18-30 8] comprising capabilities of:
 - runtime environment used for debugging (i.e., isolated environment) [col. 8, lines 8-16];
 - runtime based function used for processing debugging [col. 65, lines 24-60].

In addition, Cardoza explicitly teaches:

- a apparatus and method for testing and debugging in a computer network [abstract, col. 1, lines 5-10] comprising:

Art Unit: 2114

- status information on portion of the target operating system code or data without disturbing the operation [col. 5, lines 3-60];
- an isolated debugging environment for the computer resources (i.e., services) [col. 2, lines 35-41].

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of Applicant's invention to implement, first, to realize the You' s interactive debugging system and method in supporting the dynamic computer environment comprising capabilities of debugging services used for debugging can be executed in microprocessor, register sizes, processor register, memory addresses, operating system, hardware debugging support, operating system debugging support, software, multitude of parameters, etc... and more specifically, runtime environment used for debugging (i.e., isolated environment) and runtime based function used for processing debugging operation as being as being the maintaining an isolated debugging environment for service as claimed by Applicant since the You' s interactive debugging system and method in supporting the dynamic computer environment does perform the data tracking, checking, inspection (i.e., data monitoring), and isolating failure in supporting the continuity networking operation.

It would have further obvious to an ordinary skill in the art to keep the debugging environment to each of the service so that the debugging process can be tracked and resolved correctly without any interfering to other services; second, one would modify the You' s interactive

Art Unit: 2114

debugging system and method in supporting the dynamic computer environment to explicitly including an isolated debugging environment for the computer resources (i.e., services) as taught by Cardoza's apparatus and method for testing and debugging in a computer network in supporting the interactive debugging within a computer data networking environment.

This modification would have been obvious because a person having ordinary skill in the art would have been motivated to do so to provide the connectivity among data processor, memory, input/output device and other networking computer devices with mechanism to enhance the data processing, more specifically to interactive debugging in supporting the network establishment error detection/correction for maximizing the data process performance and throughput.

As per claims 25-41:

Due to the similarity of claims 25-41 to claims 2-24; Therefore, this claim is also rejected under the same rationale applied against claims 2-24. In addition, all of the limitations have been noted in the rejection as per claims 2-24.

As per claim 42:

Due to the similarity of claim 42 to claim 1 except for an apparatus for interactive debugging means (i.e., selecting a target means, accessing data means, monitoring data means, etc...) instead of a method for interactive debugging steps i.e., selecting a target steps, accessing data steps, monitoring data steps, etc...); Therefore, this

Art Unit: 2114

claim is also rejected under the same rationale applied against claim 1. In addition, all of the limitations have been noted in the rejection as per claim 1.

As per claim 43:

Due to the similarity of claim 43 to claim 24 except for an apparatus for multi-channel, multi-service debugging means (i.e., providing information means, maintaining an isolated debugging means, selecting a target construct for debugging means, etc...) instead of a method for multi-channel, multi-service debugging steps (i.e., providing information steps, maintaining an isolated debugging steps, selecting a target construct for debugging steps, etc...); Therefore, this claim is also rejected under the same rationale applied against claim 24. In addition, all of the limitations have been noted in the rejection as per claim 24.

As per claim 44-66:

Due to the similarity of claims 44-66 to claims 1-23 except for an apparatus for interactive debugging capabilities (i.e., a target construct, accessing data, monitoring data, etc...) instead of a method for interactive debugging steps (i.e., a target construct steps, accessing data steps, monitoring data steps, etc...); Therefore, these claims are also rejected under the same rationale applied against claims 1-23. In addition, all of the limitations have been noted in the rejection as per claims 1-23.

As per claim 67-80:

Art Unit: 2114

Due to the similarity of claims 67-80 to claims 24-43 except for an apparatus for multi-channel, multi-service debugging capabilities (i.e., providing information, maintaining an isolated debugging, selecting a target construct for debugging, etc...) instead of a method for multi-channel, multi-service debugging steps (i.e., providing information steps, maintaining an isolated debugging steps, selecting a target construct for debugging steps, etc...); Therefore, these claims are also rejected under the same rationale applied against claims 24-43. In addition, all of the limitations have been noted in the rejection as per claims 24-43.

As per claim 83:

This claim is the same as per claim 1. The only minor different is that this claim is directed to a computer readable medium instead of the method for interactive debugging steps as described in claim 1. However, it would have been obvious to one having ordinary skill in the art at the time the invention was made to realized that a computer readable medium is a necessary item for such a data processing networking system, more specifically, data debugging communication system. Since the data processing and debugging networking system obviously needs a means for instruction or code means resided within the computer readable medium for performing the data debugging, monitoring, reading, writing, storing, receiving, transmitting operation via the interactively debugging capability and via the multi-channel, multi-service debugging capabilities. Therefore, this claim is also rejected under the same rationale applied against claim 1.

Art Unit: 2114

As per claim 84:

This claim is the same as per claim 24. The only minor different is that this claim is directed to a computer readable medium instead of a method for multi-channel, multi-service debugging steps as described in claim 24. However, it would have been obvious to one having ordinary skill in the art at the time the invention was made to realized that a computer readable medium is a necessary item for such a data processing networking system, more specifically, data debugging communication system. Since the data processing and debugging networking system obviously needs a means for instruction or code means resided within the computer readable medium for performing the data debugging, monitoring, reading, writing, storing, receiving, transmitting operation via the interactively debugging capability and via the multi-channel, multi-service debugging capabilities. Therefore, this claim is also rejected under the same rationale applied against claim 24.

Response to Applicant's remarks

Applicant asserts that You in combining with Cardoza failed to teach or suggest the following:

- A. dynamically allocating a plurality of services to a plurality of processors in the multi-channel, multi-service system for execution, and selecting a target construct from the dynamically allocated executing plurality of services for debugging.
- B. maintaining an isolated debugging environment for each of the plurality of running services, wherein the isolated debugging environment provides a separate context for each

Art Unit: 2114

running service and dynamically loading one or more of the plurality of running services into the target construct.

Examiner respectfully transverses Applicant's argument as follows:

A. it is not true that the combination of both You and Cardoza failed to teach the "dynamically allocating a plurality of services to a plurality of processors in the multi-channel, multi-service system for execution, and selecting a target construct from the dynamically allocated executing plurality of services for debugging".

You explicitly teaches:

- dynamically allocating a plurality of services to a plurality of processors in the multi-channel, multi-service system for execution [col. 6, lines 20-30, and col. 87, lines 48-65];

- and interactive debugging system and method in supporting the dynamic computer environment [col. 1, lines 18-30].

You explicitly teaches capability of debugging services used for debugging can be executed in microprocessor, register sizes, processor register, memory addresses, operating system, hardware debugging support, operating system debugging support, software, multitude of parameters, etc... [col. 9, lines 16-26]. In addition, Cardoza demonstrates the apparatus and method for testing and debugging in a computer network [abstract, col. 1, lines 5-10] including an isolated debugging environment for the computer resources (i.e., services) [col. 2, lines 35-41] functionality. The combination of both You and Cardoza clearly teach the Applicant's invention since they both deal with computer's debugging process within multi-

Art Unit: 2114

services environments, such as hardware, operating system, runtime environment, target process execution, etc... [You, col. 7, lines 58-60 and col. 8, line 55-56].

- In addition, it is not true that both You and Cardoza failed to teach the capability for selecting a target construct for debugging from the plurality of running services". You's interactive debugging system and method in supporting the dynamic computer environment [col. 1, line 19] explicitly shows capabilities of cross-debugging programs from an interactive programming environment to **target execution environments** [col. 1, lines 21-23], **runtime environments, and operating system** [col. 1, lines 25-26]. You explicitly addressed target process, target program, target execution of a debugged program [col. 7, lines 50-60]. You clearly called out the "constructing of debugging object (i.e., target) within the debugging system [col. 90, lines 41-67]. Therefore, it would have been very obvious to an ordinary skill in the art to realize the You's constructing of the debugging object (i.e., target) as being the means for selecting a target construct for debugging from the plurality of running services. This is because by selecting the object or target construct for debugging within You's debugging system, the services can be easily facilitated for debugging process via inspecting, verifying, testing, [col. 2, lines 50-56], and debugging in a multi-channel, multi-service environment [col. 8, lines 55-67 and col. 79, lines 20-24].

B. First, it is not true that both You and Cardoza failed to teach the capability of "maintaining an isolated

Art Unit: 2114

debugging environment for each of the plurality of running services, wherein the isolated debugging environment provides a separate context for each running service and dynamically loading one or more of the plurality of running services into the target construct". You explicitly teaches:

- cross-debugging programs from an interactive programming environment to **target execution environments** [col. 1, lines 21-23], **runtime environments, and operating system** [col. 1, lines 25-26];
- the software/hardware exception handling policy used to allow program to stop once an exception has occurred (a breakpoint), and the target program stops [col. 86, lines 38-42];
- interrupting process used for target process [col. 86, line 53];
- breakpoint, watchpoint, exception, notification [col. 69].

In addition, you emphasized the debugging operation program includes starting, stopping, suspending, terminating, step over instructions, step into branches, step over statements, step through subroutine calls, stop at breakpoints, and stop at data watchpoints [col. 2, lines 62-67].

Therefore, it would have been obvious to an ordinary skill in the art to realize the You's software/hardware exception handling policy, interrupting process, breakpoint, watchpoint, notification, etc.. as being the maintaining an isolated debugging environment as claimed by Applicant. This is because You clearly demonstrated its

Art Unit: 2114

isolated debugging environment via these policy capabilities. It is further obvious because by applying You's exception handling policy, errors or failures within the communication/computer system can be detected, isolated, and corrected in ensuring system operation correctly.

Second, Cardoza's apparatus and method for testing and debugging in a computer network [abstract, col. 1, lines 5-10] explicitly illustrated the *isolated debugging environment for the computer resources (i.e., services)* [col. 2, lines 35-41]. Therefore, it would have been obvious to a person having ordinary skill in the art at the time of Applicant's invention to modify the You's interactive debugging system and method in supporting the dynamic computer environment to explicitly including an isolated debugging environment for the computer resources (i.e., services) as taught by Cardoza's apparatus and method for testing and debugging in a computer network in supporting the interactive debugging within a computer data networking environment.

This modification would have been obvious because a person having ordinary skill in the art would have been motivated to do so to provide the connectivity among data processor, memory, input/output device and other networking computer devices with mechanism to enhance the data processing, more specifically to interactive debugging in supporting the network establishment error detection/correction for maximizing the data process performance and throughput.

Art Unit: 2114

Conclusion

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

6. A shortened statutory period for response to this action is set to expired THREE (3) months, ZERO days from the date of this letter. Failure to respond within the period for response will cause the application to be abandoned. 35 U.S.C. 133.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dieu-Minh Le whose telephone number is (703)305-9408. The examiner can normally be reached on Monday - Thursday from 8:30 AM to 6:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert Beausoliel can be reached on (703)305-9713. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



**DIEU-MINH THAI LE
PRIMARY EXAMINER
ART UNIT 2114**

DML

6/27/04